CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method comprising:

receiving an operand at a processor;

accessing a plurality of tables to determine a floating point rounded monotonic quadratic function based on a monotonic operation;

determining a floating point result at the processor for the monotonic operation that is accurate to a unit in the last place by evaluating the monotonic quadratic function for the operand, wherein evaluating comprises indexing a plurality of tables by an index value determined by a portion of the operand, the portion comprising one third or fewer bits than the floating point result.

2. (Original) The method of Claim 1, wherein the floating point result is a 24-bit floating point result.

3. (Original) A system for evaluating a rounded arithmetic expression comprising:

a plurality of tables populated with values to generate a piecewise monotonic function;

an arithmetic unit comprising non-iterative logic coupled to the plurality of tables, the arithmetic unit comprising an input and an output, the input to receive an operand and the output to provide a floating point result for the arithmetic expression comprising an accuracy to a Unit in the Last Place (ULP);

a register comprising an input coupled to the output of the arithmetic unit.

4. (Original) The system of claim 3, wherein the arithmetic expression is a reciprocal.

5. (Original) The system of claim 3, wherein the arithmetic expression is a square root reciprocal.

6. (Original) The system of claim 3, wherein the arithmetic expression is a square root.

7. (Currently Amended) An arithmetic processor comprising:

a lookup table system including first, second and third component tables configured to provide a first operand, a second operand, a ~~fourth~~ third operand, and a square operand;

a first multiplier comprising an input to receive at least a first portion of an input operand, the first multiplier further coupled to the first component table to multiply the first operand and a square operand to provide a first result, the square operand determined responsively to the first portion of the input operand;

a second multiplier comprising an input to receive at least a second portion of the input operand, the second multiplier further coupled to the second component table to multiply the second operand and a multiplier operand to provide a second result, the multiplier operand determined responsively to the second portion of the input operand; and

an adding circuit configured to add the first result and the second result and the third operand, the third operand determined responsive to a third portion of the input operand to provide a third result;

a rounding circuit coupled to receive the third result and to provide a rounded result accurate to a unit in the last place.

8. (Original) The arithmetic processor, as recited in Claim 7, further comprising:

a square table configured to provide the square operand.

9. (Original) The arithmetic processor, as recited in Claim 8, wherein the entries in the square table are stored in a Booth recoded format.

10. (Original) The arithmetic processor, as recited in Claim 7, further comprising:

a Booth recoder configured to provide the multiplier operand.

11. (Original) The arithmetic processor, as recited in Claim 7, wherein the arithmetic processor is configured to provide a reciprocal value of an input operand.

12. (Original) The arithmetic processor, as recited in Claim 7, wherein the arithmetic processor is configured to provide a square root value of an input operand.

13. (Original) The arithmetic processor as recited in Claim 7, wherein the arithmetic processor is configured to provide a square root reciprocal value of an input operand.

14. (Original) The arithmetic processor, as recited in Claim 7, wherein the first portion of the input operand comprises high order bits of the input operand.

15. (Original) The arithmetic processor, as recited in Claim 14, wherein the first and second portions of the input operand are mutually exclusive with each other.

16. (Original) The arithmetic processor, as recited in Claim 15, wherein the first and third portions of the input operand are mutually exclusive with each other.

17. (Original) The arithmetic processor, as recited in Claim 16, wherein the second and third portions of the input operand overlap with each other.